

The Intelligent Web-Based Tutoring System using the C++ Standard Template Library

Christine Lee & *Mohd. Sapiyan Baba

KBU International College, Bandar Utama
47800 Petaling Jaya, Selangor Darul Ehsan, Malaysia
christin@kbu.edu.my

*Faculty of Computer Science and Information Technology
Universiti Malaya, 50603 Kuala Lumpur, Malaysia
pian@um.edu.my

Abstract

An intelligent tutoring system (ITS) is a computer-based instructional system with models of instructional contents that specify what to teach and teaching strategies that specify how to teach. Earlier work on creating an ITS for programming focused on the teaching programming syntax as opposed to application. The main teaching strategy is to present problem specifications for students to solve, followed by the intelligent analysis of their solutions. However, this research focused on the application level of learning programming using the C++ standard template library (STL). It has been discovered that students find C++ STL difficult due to their weaknesses in understanding various object-oriented concepts; this understanding forms the prerequisite in learning the application of the STL. As the STL is taught at the second level of a degree course, tutors face difficulties in dealing with various levels of prerequisite skills of each student. Therefore, the ITS overcomes the challenges by modelling programme specifications based on prerequisite concepts. The Bayesian theorem is applied to model the student's knowledge and direct the tutoring intelligently. This paper describes the ITS development which applies practices from the *eXtreme* programming methodology and *J2EE* technologies. An evaluation of the preliminary system testing is also included. Combining Web technologies and the ITS provides both globalisation and individualisation in teaching and learning.

INTRODUCTION

An ITS is a computer-based instructional system with models of instructional contents that specify what to teach, and teaching strategies that specify how to teach (Wenger 1987, Ohlsson 1987). Most traditional educational software focus on contents which are usually presented in a sequential manner to the user. However, different users have different needs. The main purpose of an ITS is to communicate embedded knowledge effectively and in a manner that suits the user. Therefore, the focus is not so much on the contents but on tutoring the user. How the user interacts with an ITS will, nevertheless, be influenced by the contents displayed to him/ her. More important however, the system has to be user adaptable and flexible. This is achieved through artificial intelligence (AI) techniques. There is no one right way to learn and teach the C++ language and its associated design and programming techniques. The aims and backgrounds of each student differ (Stroustrup, 1999). As the ITS seeks to mimic the human tutor while it imparts knowledge to students, it is necessary to guide students who have different levels of prerequisite skills with respect to the application of the C++ STL. The student model in an ITS has the capability of recording information on students and keeping track of their behavioural patterns as they progress in their learning. With this information, teaching and learning can proceed in a variety of ways based on their needs and interests.

MOTIVATION

Work on the ITS in the domain of C++ STL has not been recorded in available literature. Its effectiveness as a tutoring tool is accordingly the motivation for the development of this project. Empirical studies have proven that individual one-to-one tutoring is the most effective mode of teaching and learning, and this individualised tutoring is uniquely offered by the ITS (Bloom,

1984). Combining Web technologies and the ITS provides globalisation and at the same time, individualisation of teaching and learning. Globalisation in this context refers to providing similar course materials and tools to different locations in the world. This allows sharing and reusability of teaching and learning materials to save resources. In addition, institutions that franchise their courses can enhance their collaborations with their partners. On the other hand, individualisation uniquely identifies a student, providing an environment similar to face-to-face tutoring. The student actually learns in his/her own workspace in a global environment.

DOMAIN KNOWLEDGE

The STL is a framework collection of data structures (called *containers* in the STL) and algorithms designed by Stephanov & Lee (1995) at Hewlett-Packard and is based on their research on generic programming. It has been discovered that students find the C++ STL difficult due to their weaknesses in understanding various object-oriented concepts. These concepts form the prerequisite skills required in learning the C++ STL. Examples of such concepts are function overloading and parametisation. The other common problem in using the STL is the difficulty to determine the most appropriate STL algorithm to apply.

As the teaching of the C++ STL course is at a level higher than the foundation in programming, tutors teaching it cannot assume that all students have the same understanding in the elementary topics which form the prerequisites in learning the STL. Some may have already forgotten these topics and require prompting to recall them. Others may not have adequate knowledge of those topics, and hence require further instruction. Besides imparting skills on applying the STL, it is necessary for tutors to put in more effort to understand the needs of individual students. One simple and effective method to gauge a student's knowledge is through a pre-test on his/her prerequisite concepts skills.

PROPOSED DEVELOPMENT

The problems identified in the learning and teaching of the C++ STL have led to a proposed solution that models the problem specification based on prerequisite sub-skills. Tutorial questions are designed to touch on prerequisite concepts. Initially, a pre-test is conducted to gauge the student's prerequisite sub-skills. The results are then applied during the tutoring session to guide the student intelligently. The Bayesian theorem and *fuzzy* logic are applied to model his/her knowledge dynamically to provide individualised learning. The ITS also deploys various teaching strategies to enhance the learning experience.

THE SYSTEM ARCHITECTURE OF THE C++ STL ITS

Figure 1 depicts the system architecture of the C++ STL ITS which is based on a typical ITS architecture (Hartley & Sleeman, 1973). The four main modules are:

- i) The graphical user interface module
- ii) The student modelling module
- iii) The teaching strategies module
- iv) The domain knowledge module

The graphical user interface (GUI) module handles the interaction between the user (tutor or student) and the system. It accepts input from the user and directs the information to the other modules for processing. The student modelling module contains the dynamic model of a student and stores his/her personal details and knowledge. Input obtained from the GUI module through the domain knowledge module and the teaching strategies module is used to update the student model. The domain knowledge module stores tutorials, programme specifications and topics on the C++ STL. This module interacts with the teaching strategies module to allow authoring of the various teaching strategies. The responsibilities of the teaching strategies module include the authoring and maintenance of the pre-test, alternative teaching strategies and the post-test.

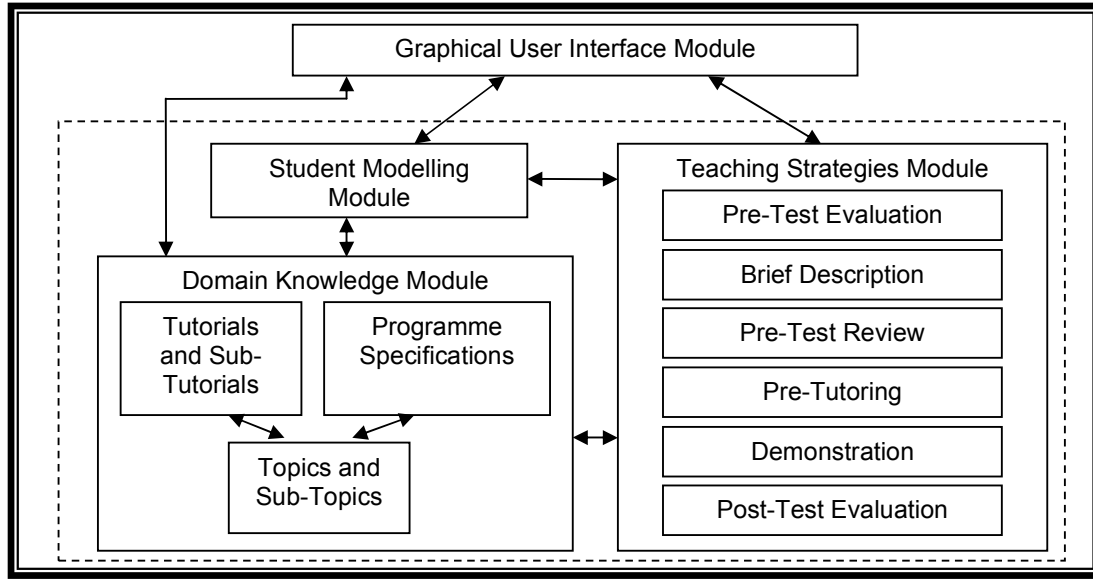


Figure 1: The System Architecture of the ITS

The Student Modelling Module

The student modelling module is responsible for building the student model from information obtained through the student's interaction with the system. The model consists of information obtained from three components. The first component is from the pre-test assessment which stores the prerequisite knowledge of the student. The second is the tutorial module which keeps track of the student's learning path during problem solving sessions and updates the student model accordingly. The third category of information for the model is derived from the post-test evaluation. This information indicates the acquired skills of the student from the tutorials completed. The student's knowledge of the domain is represented using the Bayesian theorem and categorised using fuzzy logic. Accordingly, the student model is characterised by applying a combination of a stereotype and an overlay model.

The Bayesian approach supports a transparent student model as it has a strongly proven derivation probability theory basis, and the probabilities associated are intuitively understood. To model the student's competence in the prerequisites after taking the pre-test, the desired conditional probability is that the student understands the prerequisites given that he/ she has answered the questions on them correctly. Based on an example given by Ross (2003), the conditional probability is obtained by

$$\begin{aligned}
 P(U|C) &= \frac{P(U \cap C)}{P(C)} \\
 &= \frac{P(C|U) \times P(U)}{P(C|U) \times P(U) + P(C|\sim U) \times (1 - P(U))} \quad (1)
 \end{aligned}$$

where U is the event that the student understands the prerequisite, C is the event that the student answers questions correctly. Equation (1) can be simplified to yield:

$$P(U|C) = \frac{p}{p + (1/m) \times (1 - p)} \quad (2)$$

where p is the probability that the student understands the prerequisites and m is the number of multiple choice alternatives.

The Teaching Strategies Module

The teaching strategies module consists of three main components: the pre-test, the tutorials and the post-test. The pre-test involves the setting of questions to test knowledge of the prerequisite concepts and the pre-test evaluation session. On completion of the pre-test, students are directed to the tutorials. These include a brief explanation of the required solutions and are linked to allow students to review their pre-test results vis-a-vis the problems presented in the tutorial sessions. The tutors are also presented an opportunity to design the tutorials and to set different teaching strategies. Lastly, the post-test evaluates the effectiveness of the tutorial sessions. This includes questions related to the C++ STL.

The Domain Knowledge Module

The domain knowledge is divided into three parts: topics and sub-topics, programme specifications, tutorials and sub-tutorials. The C++ STL tutor models the problem specifications based on prerequisite sub-skills. The latter are organised on a two-level hierarchical structure with topics and sub-topics. Tutorials include the programme specifications and are further divided into sub-tutorials to provide a bottom-up view of a given problem. Incorporated authoring tools allow different knowledge domains to be designed for the tests and tutorials. Therefore, the programming language is not confined to C++ but can be customised to Java, for example, and subsequently changed to the Java standard template library (JSTL).

THE EXTREME PROGRAMMING METHODOLOGY

eXtreme programming (XP) is an agile software development methodology for delivering quality software while emphasising the requirements. The application of XP in the C++ STL ITS allows the software to be implemented in a shorter period of time with tests created even before the software is written. XP contains principles and practices that guide a project development. The potential of XP is that it does small designs continuously throughout the project one task at a time (see Figure 2). Testing is performed repeatedly until the task is fully implemented and working. Refactoring is carried out to make small changes that do not affect external behaviour. It improves the design and makes the code easier to understand.

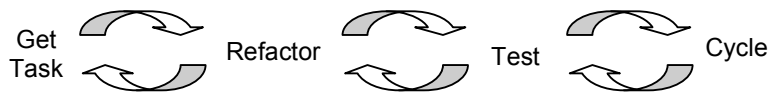


Figure 2: Cycles in XP (Astels et al., 2002)

THE THREE-TIER SYSTEM ARCHITECTURE

A great amount of research has been focused on Web-based education which gives rise to more Web-based ITS systems on the Internet. Brusilovsky (1999) gave an impressive review of adaptive and intelligent technologies in the context of Web-based distance education. He compared three categories of systems – a hybrid of adaptive hypermedia and ITS, Web-based adaptive hypermedia systems and Web-based ITS systems. Web-based ITSs first appeared during the period 1995 - 1996. Examples included RAPITS (Woods & Warren, 1995) and ELM-ART (Brusilovsky et al., 1996). This list continued to grow with ELM-ART-II (Weber & Specht, 1997), CALAT ITS (Nakabayashi et al., 1997), ILESA (López et al., 1998), AlgeBrain (Alpert et al., 1999), VC PROLOG (Peylo et al., 2000), Web PVT (Virvou & Tsiriga, 2001), WILEDS (Kassim et al., 2001), Web-based ITS using hybrid rules (Prentzas et al., 2002), SQLT-Web (Mitrovic, 2003), JITS (Sykes & Franek, 2003), BITS (Butz et al., 2004) and many more. Most of these systems use a two-tier client/server architecture implemented using Java.

The C++ STL ITS adopts a three-tier system architecture which is based on the Java 2 Platform incorporating the Enterprise Edition (J2EE) architecture. The architecture shown in Figure 3 consists of the following:

- i) The client tier layer
- ii) The middle tier layer
- iii) The data source tier layer

The combination of XP and J2EE forms a strong foundation in the Web development of the C++ STL ITS. Their industry practices are applied to enhance the value and contribution of this research. It is an attempt to bring the C++ STL ITS closer to the industry level and promote its adoption.

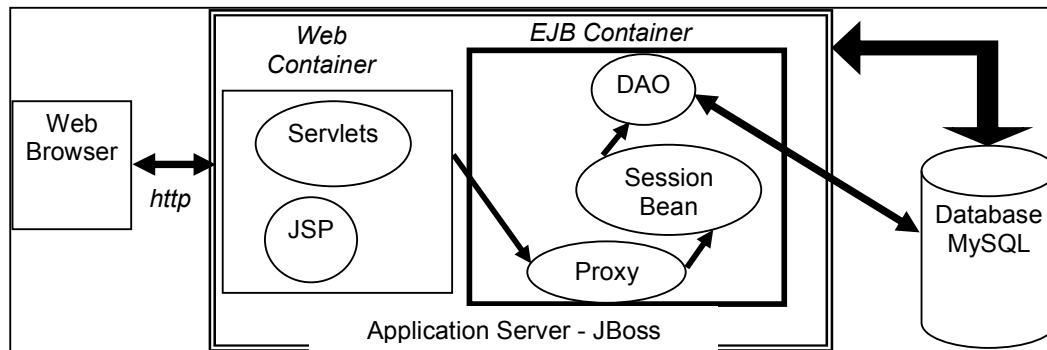


Figure 3: The Three-tier System Architecture of C++ STL ITS

The Client Tier Layer

The client tier which represents the graphical user interface module in the ITS consists of two parts:

- i) Dynamic Web pages containing HTML, which are generated by the Web container components running in the middle tier.
- ii) A Web browser, which reads the pages received from the application server and interacts with the user.

The Middle Tier Layer

The middle tier accepts user responses from the client tier and generates the appropriate presentation logic. This tier also handles the core business logic of the application implemented as Enterprise Java Bean (EJB) components. It includes the following:

- The application server
- The Java servlet and servlet page server
- EJB components (consisting of data access objects (DAO), session beans and proxy)

The Data Source Tier Layer

The student module, the teaching strategies module and the domain knowledge module reside in the data source tier layer. At this tier, the DAO is used to invoke the MySQL database.

SYSTEM EVALUATION

One of the primary concerns of the C++ STL ITS is to ascertain the prerequisite knowledge of the students after the pre-test. The objective of the test was to find out the applicability of the Bayesian theorem to model the student's understanding. Therefore, on completion of the development, the preliminary system testing was carried out on the pre-test. Figures 4 and 5

depict two main screen shots of the C++ STL ITS. Figure 4 shows an example of a pre-test question, whereas Figure 5 depicts a tutorial session. Each tutorial was divided into sub-tutorials and students were presented with different levels of teaching strategies (or hints) based on their pre-test results.

The preliminary testing was conducted for the pre-test with 34 students as the participants. There were all together 70 multiple choice single answer questions in the pre-test covering 12 main prerequisite topics – fundamentals, selection, iteration, array, user defined function, class, class member function, constructor, function template, class template, operator overloading and class string. The conditional probabilities of the students' understanding in the prerequisites were calculated at the end of the pre-test. Table 1 summarises the overall performance for each topic. It is consistent with the fact that students are better at fundamental topics compared to object-oriented topics and templates. From the average values, a threshold of 0.80 is able to show a student's mastery of a prerequisite concept. A value below 0.60 indicates a student's weakness. Using these thresholds, students can be directed to different remedial lessons during the tutorial session.

Gurney, Intelligent Tutoring System (ITS) Ver. BETA 1.0
Login Time: 2005-08-12 15:36:11.312
ST-001

[Home](#) | [Pre-Test](#) | [Tutorial](#) | [Results](#)
[Logout](#)

Question 26.

Accessing Members Question 1

Given the following declaration of a member function in the class Dog:
`void setBreed(string aBreed);`

write the statement to call the function aDog object.

Available Answers

Answer	Choice
<code>aDog.setBreed('German Shepherd');</code>	<input type="radio"/>
<code>aDog.setBreed("German Shepherd");</code>	<input type="radio"/>
<code>aDog.setBreed(German Shepherd);</code>	<input type="radio"/>
<code>setBreed.aDog("German Shepherd");</code>	<input type="radio"/>

[Next](#)
[Question Contents](#)

Figure 4: A Sample Pre-test Question

Table 1: Overall Students' Pre-test Performance for Each Topic

TOPIC	TOTAL	AVERAGE OF P(U C)
Class String	27.94	0.82
Operator Overloading	26.40	0.78
Class Template	23.00	0.68
Function Template	28.85	0.85
Constructor	28.23	0.83
Class Member Function	29.03	0.85
Class	29.35	0.86
User Defined Function	30.05	0.88
Array	31.94	0.94
Iteration	31.20	0.92
Selection	32.68	0.96
Fundamentals	32.34	0.95

Title : STL vector Question 1

Please analyze the tutorial framework below and answer each related sub-tutorial questions.

```
#include <iostream>
#include <>

using namespace std;

int main()
{
    // Declare a vector of 10 integers, myVector
        
    int i;
```

Sub Tutorials

Sub-Tutorial 1

Sub-Tutorial 2

Sub-Tutorial 3

Write the statement to create a vector called myVector that can store up to 10 integers.

Hint ?

Submit All Answers

Figure 5: A Screen Shot of a Tutorial Session

CONCLUSION

The conditional probabilities obtained from the pre-test reflected the students' understanding of a particular prerequisite. It provided a form of automated self-assessment for the students as it indicated their competency in each topic. The application of the Bayesian theorem resulted in information on the students' prerequisite knowledge. These results, with the thresholds identified, were useful in directing the students during the tutoring sessions, as the student model could then be updated based on the initial values obtained. The objective of the testing was been achieved and the results shown to be beneficial for the tutoring sessions as the students could then be guided intelligently.

REFERENCES

- Alpert, S.R., Singley, M.K. & Fairweather, P.G. (1999). Deploying intelligent tutors on the web: an architecture and an example. *International Journal of Artificial Intelligence in Education*, 10(2), 183-197.
- Astels, D., Miller, G. & Novak, M. (2002). *A Practical Guide to eXtreme Programming*. Upper Saddle River, NJ, USA: Prentice Hall.
- Bloom, B.S. (1984). The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4-16.
- Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education. In C. Rollinger & C. Peylo (Eds.). *Special Issue on Intelligent Systems and Teleteaching*, 4, 19-25. Künstliche Intelligenz.
- Brusilovsky, P., Schwarz, E. & Weber, G. (1996). ELM-ART: An intelligent tutoring system on world wide web. Proceedings of the *ITS'96*, 261-269.
- Butz, C.J., S. Hua, S. & Maguire, R.B. (2004). A Web-based intelligent tutoring system for computer programming. In *IEEE/WTC/ACM Conference on Web Intelligence (WI04)*, 159-165.
- Hartley, J. & Sleeman, D. (1973). Towards more intelligent teaching systems. *International Journal of Man-Machine Studies*, 2, 215-236.
- Kassim, A.A., Ahmed, K.S. & Ranganath, S. (2001). A Web-based intelligent approach to tutoring. Paper presented at the International Conference on Engineering Education, Session 8B4, Oslo, Norway, 6-10 August.
- López, J. M., Millán, E., Pérez-de-la-Cruz, J. L. & Triguero, F. (1998). ILESA: A web-based intelligent learning environment for the simplex algorithm. In Alvegård, C. (ed.) Proceedings of *CALISCE'98*, 4th International Conference on Computer Aided Learning and Instruction in Science and Engineering, Göteborg, Sweden, 399-406.
- Mitrovic, A. (2003). An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4), 173-197.
- Nakabayashi, K., Maruyama, M., Koike, Y., Kato, Y., Touhei, H. & Fukuhara, Y. (1997). Architecture of an intelligent tutoring system on the WWW. In Proceedings of the *8th World Conference of the AIED Society*, Kobe, Japan, 18-22 August.
- Ohlsson, S. (1987). Some principles of intelligent tutoring. In Lawler & Yazdani (eds.), *Artificial Intelligence and Education*, 1, 203-238, Norwood, NJ: Ablex.
- Peylo, C., Thelen, T., Rollinger, C. & Gust, H. (2000). A Web-based intelligent educational

system for PROLOG. In Peylo, C. (Ed.). Proceedings of the *International Workshop on Adaptive and Intelligent Web-based Educational Systems*, Technical Report of the Institute for Semantic Information Processing, Osnabrück, 85-96.

Prentzas, J., Hatzilygeroudis, I. & Garofalakis, J. (2002). A web-based intelligent tutoring system using hybrid rules as its representational basis. In Proceedings of the *ITS-2002*, Biarritz, France, June, 119-128.

Ross, M.S. (2003). *Introduction to Probability Models*. 8th Edition, USA: Academic Press.

Stephanov, A. & Lee, M. (1995). The standard template library.
<http://butler.hpl.hp.com/stl>

Stroustrup, B. (1999). Learning standard C++ as a new language, AT&T Labs, *The C/C++ Users Journal*, May, 43-54.

Sykes, E. R & Franek, F. (2003). An intelligent tutoring system prototype for learning to program java. Proceedings of the *3rd IEEE International Conference on Advanced Learning Technologies (ICALT '03)*, Athens, Greece, July, 485-486.

Virvou, M. & Tsiriga, V. (2001). Adaptive tutoring based on the student model of a web-based ICALL. Proceedings of the *TELEMATICA-2001 International Conference on Telematics and Web Based Education*, 47-50.

Weber, G. & Specht, M. (1997). User modeling and adaptive navigation support in WWW-Based tutoring system. In A. Jameson, C. Paris, & C. Tasso (Eds.) *User Modeling: Proceedings of the Sixth International Conference, UM97*. Vienna, New York: Springer Wien New York, 289-300.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.

Woods, P. & Warren, J. (1995). Rapid prototyping of an intelligent tutorial system. Proceedings of the *ASCILITE'95*, Melbourne, 557-563.